**IN THE SPECIFICATION**

Please respond to the informality objections as follows.

Please amend paragraph 5 on page 2 as follows:

To avoid this linearity problem, a combining function, whose inputs are taken from the outputs of several LFSRs in parallel, is used to destroy the linearity of the original sequence generated according to the LFSRs. In convention, the combining function employed is a fixed function. Therefore, the mapping defined by the combining function is a one-to-one mapping, and for the same input imposed on the combining function, the same ~~input~~ output will be obtained. Such a generator suffers a divide-and-conquer attack if a correlation exists between the pseudo-random sequence and the output sequence of individual LFSRs. One solution could be to use the Data Encryption Standard (DES) to randomize the output but this is not economical as a substantial amount of hardware is required.

Please amend paragraph 33 on page 8 as follows:

Referring to Figure 2, the periodic Sequence Generator 11 consists of n Linear Feedback Shift Registers (LFSRS) is providing a n-bit output N. In the preferred embodiment there are 12 LFSRs providing a 12-bit output N. The general structure of the LFSRs 18 is shown in Figure 3. In the preferred embodiment the LFSRs 18 have n elements S, but alternative embodiments may have more or less elements as will be apparent to the skilled addresses. The length of the n LFSRs 18 is ~~are~~ pairwise relatively prime such that the output of the Periodic Sequence Generator 11 has a period of $\prod_i^n(2^{L_i}-1)$ where $L_i$ is the length of the i LFSR. The initial contents of the LFSRs 18 elements S are filled with a secret key for the Pseudo-random Sequence Generator. The n-bit output N of the Periodic Sequence Generator 11 goes into the Function Generators 12 & 13.